T. N. E. Greville
Theory and Applications
of Spline Functions -
(A Albert: the Theory of
splines in math. in sc. and eng.)
↑ TARONI

Some Algorithms for the Computation of Interpolating and Approximating Spline Functions

L. L. SCHUMAKER

1. Introduction. Numerous papers developing the theory of interpolation and approximation by spline functions have appeared in recent years. Concurrent with this theoretical development, there has been considerable interest in practical algorithms for the computation of splines. Unfortunately, however, a great deal of the work on algorithms has not been documented, or at best is available only in local reports of the academic departments, and industrial, government, or private laboratories in which it was carried out (cf. the references cited at the end of this article). Thus, a great wealth of computational experience and accompanying insight is not so readily available to the prospective user of spline methods.

The purpose of the present survey is to at least make the reader aware of the existence of various algorithms for computing interpolating and approximating splines, to outline their derivation whenever practicable, and to make some attempt to discuss the numerical experience obtained with them. Because of the above-mentioned relative inaccessibility of sources, it is essential that a strong disclaimer of any kind of completeness be made at the outset.

The first part of the paper is devoted to some typical interpolation problems and corresponding algorithms for their solutions, while the second part is concerned with approximation methods. Whenever possible, examples of the kinds of problems the algorithms were successfully tested on are quoted, as well as some indications of where failures occurred. The details of programming the algorithms are completely ignored, and no existence, uniqueness or convergence results are established in this survey. The reader is strongly advised to

consult the available references as well as experienced practitioners if possible before attempting large scale computational use of what appears here.

2. Interpolating splines.

2.1 Interpolation at nodes differing from the knots. A spline function of degree n with k prescribed knots $x_1 < x_2 < \ldots < x_k$ is a function $s(t) \in C^{n-1}$ such that $s(t) \in \pi_n$, the class of polynomials of degree at most n, in each of the intervals $(-\infty, x_1), (x_1, x_2), \ldots, (x_k, \infty)$. Denote by $S_{n,k}(x_1, \ldots, x_k)$ this class of splines.

Problem 2.1 Given knots $x_1 < x_2 < \ldots < x_k$, points $t_1 < t_2 < \ldots < t_{n+k+1}$ satisfying

$$t_i < x_i < t_{n+i+1}, \qquad i = 1, 2, \ldots, k \tag{2.1}$$

and real numbers $\{y_i\}_1^{n+k+1}$, compute a spline function of degree n with knots $\{x_i\}_1^k$ interpolating $\{y_i\}_1^{n+k+1}$ at the $\{t_i\}_1^{n+k+1}$.

We first remark that since $\{1, t, \ldots, t^n, (t - x_1)_+^n, \ldots, (t - x_k)_+^n\}$ form a basis for $S_{n,k}(x_1, \ldots, x_k)$, (cf. [15]), s must satisfy

$$s(t_j) = \sum_{i=0}^{n} a_i t_j^i + \sum_{i=1}^{k} c_i (t_j - x_i)_+^n = y_j, \qquad j = 1, 2, \ldots, n+k+1. \tag{2.2}$$

Since the determinant of (2.2) is nonzero (see Lemma 2.2 of [22]), s exists uniquely.

The simplest possible algorithm for computing s would amount to solving the linear system (2.2) by any of the usual methods. This turns out to be impossible in practice, however, since the matrix of the system is very badly conditioned unless n and k are both small. To derive a viable algorithm for obtaining s numerically it is necessary to find a different basis for $S_{n,k}(x_1, \ldots, x_k)$.

Choose $x_{-n} < x_{-n+1} < \ldots < x_0 < x_1$ and $x_k < x_{k+1} < \ldots < x_{k+n+1}$. For $i = 1, 2, \ldots, n+k+1$ define the B-splines

$$(2.3) \qquad M_i(t) = M[t; x_{i-n-1}, x_{i-n}, \ldots, x_i]$$

where $M[\cdot]$ denotes the $(n+1)$th divided difference on the points $x_{i-n-1}, x_{i-n}, \cdots x_i$ of the function $M(t;y) = (n+1)(y-t)_+^n$. By Theorem 5 of Curry and Schoenberg [6], every $s(t) \in \mathcal{S}_{n,k}(x_1, \ldots, x_k)$ has a unique expansion

$$(2.4) \qquad s(t) = \sum_{i=1}^{n+k+1} c_i M_i(t) \quad \text{for } x_0 \leq t \leq x_{k+1}.$$

This basis for $\mathcal{S}_{n,k}(x_1, \ldots, x_k)$ leads to the following convenient interpolation method.

Algorithm 2.2 (Compute the coefficients $\{c_i\}_1^{n+k+1}$ in the expansion (2.4) for the solution of Problem 2.1).

(a) Set $h = (x_k - x_1)/(k-1)$, $x_0 = \min(t_1, x_1 - h)$,

$x_{k+1} = \max(t_{n+k+1}, x_k + h)$, and $x_{-i} = x_0 - ih$

$x_{i+k+1} = x_{k+1} + ih$ for $i = 1, 2, \ldots, n$.

(b) Solve the system

$$(2.5) \qquad \sum_{i=1}^{n+k+1} c_i M_i(t_j) = y_j, \qquad j = 1, 2, \ldots, n+k+1$$

for the $\{c_i\}_1^{n+k+1}$.

This particular interpolation problem has been all but neglected, and this seems to be the first time an algorithm concerned with it has appeared in print. The method is easily modified to allow the computation of Tchebycheffian splines (see §6 of [22]) interpolating as in Problem 2.1.

Algorithm 2.2 has been programmed in Algol at the Mathematics Research Center for the Burroughs 5500 computer, and has been tested on a number of examples. Good results for fairly wild data (y_i's oscillating up to say 10^4) have been obtained for modest sizes in the degree of the splines, e.g., up to 10. For high degrees some difficulty in accurately computing the divided differences involved in obtaining the $M_i(t_j)$ was encountered. It should be remarked that the matrix in (2.5) is of the "banded" type because of the restricted support of the M_i's, and this local property of the expansion

is also favorable in evaluating the spline.

2.2 Natural splines. A function $s(t)$ is called a natural spline of degree $2m-1$ with knots $x_1 < x_2 < \ldots < x_k$ provided $s \in S_{2m-1,k}(x_1, \ldots, x_k)$ (cf. § 2.1) and $s(t) \in \pi_{m-1}$ in each of the intervals $(-\infty, x_1)$, (x_k, ∞).

Problem 2.3 Given $x_1 < x_2 < \ldots < x_k$ and real numbers $\{y_i\}_1^k$ compute a natural spline function of degree $2m-1$ interpolating $\{y_i\}_1^k$ at the $\{x_i\}_1^k$.

The existence of a unique s satisfying Problem 2.3 ($m \leq k$) follows from the fact that it is determined by the system of equations (cf [15])

$$(2.6)\qquad \begin{cases} s(x_j) = \sum_{i=0}^{m-1} a_i x_j^i + \sum_{i=1}^{k} \lambda_i (x_j - x_i)_+^{2m-1} = y_j, \\ \qquad\qquad\qquad\qquad\qquad\qquad j = 1, 2, \ldots, k \\ \sum_{i=1}^{k} c_i x_i^r = 0 \qquad\qquad r = 0, 1, \ldots, m-1. \end{cases}$$

As in §2.1, the simplest algorithm for solving Problem 2.3 would involve numerical solution of (2.6), but the same objection applies; it is an ill-conditioned system. Several alternate approaches have been devised.

Let $n = 2m-1$ and choose $\{x_i\}_{-n}^{0}$ and $\{x_i\}_{k+1}^{n+k+1}$ as $x_{1-i} = x_1 - ih$, $x_{i+k} = x_k + ih$, $i = 1, 2, \ldots, n+1$, where $h = \frac{x_k - x_1}{k-1}$. Let $\{M_i(t)\}_1^{n+k+1}$ be the B-splines defined in (2.3), and let (2.4) be the expansion of s. Now

$$(2.7)\qquad s(x_j) = \sum_{i=1}^{n+k+1} c_i M_i(x_j) = y_j, \qquad j = 1, 2, \ldots, k$$

provide k equations for the determination of the $\{c_i\}_1^{n+k+1}$. Consider dividing the intervals $[x_0, x_1]$ and $[x_k, x_{k+1}]$ into n equal parts. Since $s^{(m)}(t) = 0$ in these intervals we have

$$(2.8)\qquad \begin{cases} \Delta^m s(x_0 + j h_0) = 0 \qquad j = 0, 1, \ldots, m-1 \\ \Delta^m s(x_k + j h_1) = 0 \qquad j = 0, 1, \ldots, m-1 \end{cases}$$

where $h_0 = (x_1 - x_0)/n$, $h_1 = (x_{k+1} - x_k)/n$, and the differences are taken with respect to j . The equations (2.7) combined with (2. 8) provide $n+k+1$ equations for the $n+k+1$ c_i's .

Algorithm 2. 4 (Determine coefficients $\{c_i\}_1^{n+k+1}$ in the expansion (2. 4) of s solving Problem 2. 3).

a) Set $x_{1-i} = x_1 - ih$ and $x_{i+k} = x_k + ih$ for $i = 1, 2, \ldots, n+1$, $h = (x_k - x_1)/(k - 1)$.

b) Compute $M_i(x_j)$, $i=1, 2, \ldots, n+k+1$, $j=1, 2, \ldots, k$.

c) Compute $\Delta^m s(x_0 + j h_0)$ $j = 0, 1, \ldots, m-1$.

$\Delta^m s(x_k + j h_0)$ $h_0 = h/n$.

d) Solve system (2.7), (2. 8) for $\{c_i\}_1^{n+k+1}$.

Algorithm 2. 4 has also been tested at the Mathematics Research Center on a number of examples, again with fairly wild data. As with Algorithm 2. 2 good results were obtained modulo some difficulty in computing the divided differences for higher degree splines. The matrix here is also fairly sparse, and the local property of the basis likewise obtains.

Still another method for computing the solution to Problem 2. 3 relies on obtaining an expression for $s^{(m)}(t)$. We need to introduce the notation

$$D\begin{pmatrix} \varphi_1, \varphi_2, \ldots, \varphi_m \\ z_1, z_2, \ldots, z_m \end{pmatrix} = \sum_{i=1}^{m} \frac{\varphi_i}{(z_i - z_1)\ldots(z_i - z_{i-1})(z_i - z_{i+1})\ldots(z_i - z_m)}$$

for the m-th divided difference of the data $\varphi_1, \varphi_2, \ldots, \varphi_m$ at the points $z_1, z_2, \ldots, z_m$. It can be shown that

$$(2.9) \qquad s(t) = \sum_{i=1}^{k-m} \frac{\lambda_i M_i^*(t)}{(2m)!} + \sum_{i=0}^{m-1} q_i t^i$$

where

$$(2.10) \qquad M_i^*(t) = 2m\, D\begin{pmatrix} (x_i - t)_+^{2m-1}, & \ldots, & (x_{i+m} - t)_+^{2m-1} \\ x_i, & \ldots, & x_{i+m} \end{pmatrix}$$

$$(2.11)\qquad \sum_{i=0}^{m-1} q_i x_j^i = y_j - \sum_{i=1}^{k-m} \frac{\lambda_i M_i^*(x_j)}{(2m)!}, \quad j=1, 2, \ldots, m,$$

$$(2.12)\qquad \sum_{i=1}^{k-m} \lambda_i A_{ij} = D\begin{pmatrix} y_j, y_{j+1}, \ldots, y_{j+m} \\ x_j, x_{j+1}, \ldots, x_{j+m} \end{pmatrix}, \quad j=1, 2, \ldots, k-m$$

and

$$(2.13)\qquad A_{ij} = \frac{1}{(2m)!} D\begin{pmatrix} M_i^*(x_j), \ldots, M_i^*(x_{j+m}) \\ x_j \quad , \ldots, x_{j+m} \end{pmatrix}.$$

Algorithm 2.5 (Determine $\{\lambda_i\}_1^{k-m}$, $\{q_i\}_0^{m-1}$ in the expression (2.9) for the solution of Problem 2.3.)

a) Evaluate $D\begin{pmatrix} y_j, \ldots, y_{j+m} \\ x_j, \ldots, x_{j+m} \end{pmatrix}$, $j=1, 2, \ldots, k-m$.

Evaluate $M_j^*(x_i)$, $i=1, 2, \ldots, k$, $j=1, 2, \ldots, k-m$.

Evaluate A_{ij} by (2.13) $i, j=1, 2, \ldots, k-m$.

b) Solve (2.12) for $\{\lambda_i\}_1^{k-m}$.

c) Solve (2.11) for $\{q_i\}_0^{m-1}$.

It may be noted that step c) amounts to determination of the unique interpolating polynomial of degree $m-1$ for m given data points.

This algorithm has been tested on the same data as Algorithm 2.4, with comparable results, although because of the nonlocal character of the underlying basis, some loss in accuracy near the right-hand end of the interval was noticed for large numbers of knots. Moreover, accuracy was found to suffer when extremely nonuniform knot spacings were prescribed, e.g., with the ratio of largest to smallest knot distance on the order of 100 or 1000.

Numerous authors have obtained Algorithm 2.5, or slight modifications of it, see, e.g., [1, 2, 3, 4, 5, 6, 11, 14, 16] etc. The above form was taken from [16], where it is shown to also be applicable for Tchebycheffian splines (see §6 of [22]). Considerably more general splines have been studied, and analogous algorithms for their

computation are available, see [17] and references therein.

The idea of a smoothing spline (see [18, 19]) is very closely related to interpolating splines, and there is an algorithm in the literature [18] for computing them in the cubic case.

3. Uniform approximation.

3.1 Fixed knots. Let $a < x_1 < x_2 < \ldots < x_k < b$ be prescribed and let $\mathcal{S}_{n,k}(x_1, \ldots, x_k)$ be the class of splines of degree n with knots $\{x_1\}_1^k$ defined in §2.1.

Problem 3.1. Given $f \in C[a, b]$, find $s^*(t) \in \mathcal{S}_{n,k}(x_1, \ldots, x_k)$ such that

$$\|f - s^*\|_\infty = \max_{a \le t \le b} |f(t) - s^*(t)| \le \|f - s\|_\infty$$

for every $s \in \mathcal{S}_{n,k}(x_1, \ldots, x_k)$.

The theory of this best approximation problem has been surveyed in [22] in this volume. In particular, we recall that a best approximation exists although in general it may not be unique.

For computational purposes, we are forced to replace Problem 3.1 with the following discretized version.

Problem 3.2. Let N be a positive integer and let $D = \{a + ih\}_0^N$, where $h = (b-a)/N$. Given $f \in C[a, b]$ find $s^* \in \mathcal{S}_{n,k}(x_1, \ldots, x_k)$ such that

$$\|f - s^*\|_D = \max_{x \in D} |f(x) - s^*(x)| \le \|f - s\|_D$$

for every $s \in \mathcal{S}_{n,k}(x_1, \ldots, x_k)$.

The existence of best approximations in this sense is also easily verified, and most of the characterization theory quoted in [22] also carries over. Of special interest is the following de la Vallée Poussin type of result for splines (cf. Theorem 2.8 of [22]).

Theorem 3.3. Suppose there exist $\{t_i\}_1^{n+k+2} \subset D$ and $\{\lambda_i\}_1^{n+k+2}$ with $\lambda_1 > 0$ and $a \le t_1 < t_2 < \ldots < t_{n+k+2} \le b$ satisfying

$$(3.1) \qquad (f - s)(t_i) = (-1)^i \xi \lambda_i, \qquad i=1, 2, \ldots, n+k+2 \ (\xi = \pm 1).$$

Then

$$\min_i \lambda_i \leq A_{n,k}(D) \leq \|f - s\|_D ,$$

where

$$A_{n,k}(D) = \min_{s \in \mathcal{S}_{n,k}(x_1, \ldots, x_k)} \|f - s\|_D .$$

The importance of this result here, is that if some s exists satisfying (3.1) then upper and lower bounds for the deviation $A_{n,k}(D)$ are immediately available. For certain choices of $\{t_i\}_1^{n+k+2}$ the construction of such splines is quite easy. We call a set $M_0 = \{t_i^{(0)}\}_1^{n+k+2} \subset D$ satisfying

$$(3.2) \qquad a \leq t_1^{(0)} < t_2^{(0)} < \ldots < t_{n+k+2}^{(0)} \leq b$$

admissible provided there exists a unique spline $s_0(t) \in \mathcal{S}_{n,k}(x_1, \ldots, x_k)$ such that

$$(3.3) \qquad s_0(t_j^{(0)}) + (-1)^j \lambda_0 = f(t_j^{(0)}), \quad j=1, 2, \ldots, n+k+2,$$

for some $\lambda_0 \neq 0$; it is shown in [21] that a set $M = \{t_i^0\}_1^{n+k+2}$ is admissible provided there exist $\{\gamma_i\}_1^{n+k+1} \subset M$ such that $\gamma_i < x_i < \gamma_{n+i+1}$, $i = 1, 2, \ldots, k$. Now, as mentioned above, $\mathcal{S}_{n,k}(x_1, \ldots, x_k)$ is a finite-dimensional linear space, and hence if $\{b_i(t)\}_1^{n+k+1}$ provide a basis for it, then we may seek s satisfying (3.3) in the form $s(t) = \sum_{i=1}^{n+k+1} a_i b_i(t)$, i.e., we solve the system of equations

$$(3.4) \qquad \sum_{i=1}^{n+k+1} a_i b_i(t_j^0) + (-1)^i \lambda_0 = f(t_j^{(0)}) ,$$

$j=1, 2, \ldots, n+k+2$, for $\{a_i\}_1^{n+k+1}$ and λ_0.

These ideas were exploited in [21] to derive a Remez type single-exchange method wherein M is adjusted so that larger and larger λ's are obtained thus providing a sharp lower bound for $A_{n,k}(D) \leq A_{n,k} = \min_{s \in \mathcal{S}_{n,k}(x_1, \ldots, x_k)} \|f-s\|_\infty$.

An upper estimate for $A_{n,k}$ is provided by $\|f - s\|_D + \omega(h)$, where ω is the modulus of continuity of f.

Since at each step of this algorithm the linear system (3.4) must be solved, it is important to choose a good basis for $S_{n,k}(x_1, \ldots, x_k)$ which avoids ill-conditioned matrices. It has been my experience that the basis exhibited in (2.3) provides excellent results. Convergence properties of this algorithm have not been established, but it has been shown that the sequence of λ's produced are increasing (see [21]).

An alternate attack on Problem 3.2 due to Esch and Eastman [9] consists in considering it as a linear programming problem. Standard computational techniques for such problems can then be applied. The linear programming method is recommended since the Remez algorithm discussed above is subject to restrictions which can cause it to terminate prior to obtaining the best approximation with respect to $\|\cdot\|_D$ (see [21]).

These algorithms have been tested on a number of smooth and unsmooth functions. For purposes of comparison we include an example in which we approximate the roof function

$$r(t) = \begin{cases} t & \text{if } t \le 1/2 \\ 1-t & \text{otherwise} \end{cases}$$

on $[0,1]$ with $N = 100$ and $D = \{i/100\}_0^{100}$:

n	k	$A_{n,k}(D)$	Knot Locations
1	1	0	.5
9	0	1.7×10^{-2}	
6	0	2.4×10^{-2}	
3	1	2.7×10^{-2}	.500
3	3	1.5×10^{-4}	.499, .500, .501
2	2	9.1×10^{-4}	.499, .500

The following general remarks seem warranted by the experiments: (1) for smooth functions, splines in $\mathcal{S}_{n,k}$ do not do appreciably better than polynomials with the same number of parameters (i.e., n+k+1); (2) for less smooth functions which are more or less piecewise smooth, splines are by far superior provided the knots are properly located.

3.2 Free knots. In this section the class of approximating splines will be taken to be

$$\mathcal{S}_{n,k} = \{s(t) \mid \text{there exist } a = x_0 < x_1 < \ldots < x_{r+1} = b \text{ and integers } m_1, \ldots, m_r \text{ with } 1 \le m_i \le n+1 \text{ and } \sum_{i=1}^{r} m_i = k \text{ such that } s(t) \in \pi_n \text{ in each of the intervals } (x_i, x_{i+1}) \text{ while } s \in C^{n-m_i} \text{ in an open neighborhood of } x_i,\ i = 1, 2, \ldots, r\}.$$

This is the class of all splines of degree n with some k knots in the interval [a, b] allowing multiple knots up to multiplicity n+1 (see [22], §§2, 3).

Problem 3.4. Given $f \in C[a, b]$ find $s^* \in \mathcal{S}_{n,k}$ such that

$$\|f - s^*\|_\infty = \max_{a \le t \le b} |f(t) - s^*(t)| \le \|f - s\|_\infty$$

for every $s \in \mathcal{S}_{n,k}$.

Theoretical results for this problem have been surveyed in [22]. As with Problem 3.1, in practice one must replace Problem 3.4 with a discretized version as in Problem 3.2. Research on effective methods for obtaining best spline approximations with free knots is still in progress. Esch and Eastman [9, 10] have reported quite successful experience with a method based on solving the fixed knot problem coupled with a knot adjustment procedure. Their method is predicated on the assumption that there exists a best approximating $s \in \mathcal{S}_{n,k}$ such that $f - s$ exhibits a normal alternating character, i.e., there exist $a \le t_1 < \ldots < t_{n+k+2} \le b$ such that

$$(f - s)(t_i) = (-1)^i \zeta B_{n,k}, \quad i = 1, 2, \ldots, n+k+2 \ (\zeta = \pm 1),$$

where $B_{n,k} = \min_{s \in \mathcal{S}_{n,k}} \|f - s\|_\infty$. This, however,

remains a conjecture (cf. discussion in §3.2 of [22]). All of the examples treated in [9, 10] possessed alternating best approximations, but on the other hand, they also satisfied $f^{(n+1)}(t) \geq 0$ which is known to be sufficient to guarantee the normal alternating characterization (see Theorem 5.2 and the discussion in §5.5 of [22]).

Since the details of the Esch and Eastman algorithm are rather complicated, we do not include them here. We shall, however, quote a few of their examples and discuss some of their computational experience.

The first example is approximation of the analytic function $f(x) = e^x$ on $[0,1]$ (see Table 1 below). It is seen that increasing the degree of the spline was more effective than adding knots.

The second example involves approximating $\sqrt{x}$ on $[0,1]$ (Table 2). Since $\sqrt{x}$ is not so well behaved near 0, it is perhaps not surprising that the splines were more effective than polynomials with a comparable number of parameters.

4. L_2 approximation.

4.1 Fixed knots. For prescribed $a < x_1 < x_2 < \ldots < x_k < b$ let $S_{n,k}(x_1, \ldots, x_k)$ be as in §§2.1, 3.2.

Problem 4.1. Given $f \in C[a,b]$ determine $s^* \in S_{n,k}(x_1, \ldots, x_k)$ such that

$$\|f - s^*\|_2 = \int_a^b [f(t) - s^*(t)]^2 \, dt \leq \|f - s\|_2$$

for every $s \in S_{n,k}(x_1, \ldots, x_k)$.

Very little theory is available for best L_2 approximations (see §4 of [22], pp. 65-85, this volume). Since $S_{n,k}(x_1, \ldots, x_k)$ has dimension $n+k+1$, the computational problem is a standard least squares problem, i.e., given a basis $\{b_i(t)\}_1^{n+k+1}$, determine an orthonormal basis $\{\psi_i(t)\}_1^{n+k+1}$, e.g., by Gram-Schmidt. The choice of initial basis and orthonormalization procedure are critical, however. A complete discussion as well as an explicit algorithm (and program) can be found in de Boor and Rice [7]. It has been tested extensively and has provided smooth spline approximations in a variety of problems.

k = Number of Knots

n = Degree of Splines

	0	1	2	3	4	5	6	7
0	.859141	.429570		.214785				.107393
1	.105933							
2	.008756	.001342 .5429	.000427 .3809 .6951	.00018701 .2934 .5418 .7713	.00009789 .2386 .4440 .6363,.8170	.00005750 .2011 .3762,.5416 .6984,.8475	.00003659 .1737,.3263 .4715,.6100 .7424,.8692	.00002472 .1530,.2882 .4175,.5416 .6607,.7752 .8856
3	.00054478	.00006396 .5322	.00001597 .3833 .6748	.00000572 .2996 .5315 .7507	.00000253 .2458 .4385 .6222,.7979	.00000128 .2085 .3733,.5314 .6836,.8302		
4	.00002716	.00000255 .5258	.00000052 .388 .660	.00000016 .308 .527 .736				
5	.00000113	.00000009 .523	.00000002 .41 .67					
6	.00000004							

TABLE I

Optimal-Knot Spline Approximations

to $f(x) = e^x$ on $[0,1]$.

The first entry in each box is the minimax error magnitude. Subsequent entries give the optimal knot locations.

k = Number of Knots

n = Degree of Spline

	0	1	2	3
0	.500000	.250000 .2500	.166667 .1111 .4444	.125000 .0625 .2500 .5625
1	.125000	.041667 .1111		
2	.067619	.019158 .0590	.008226 .01088 .1523	.004347 .00305 .04144 .2381
3	.045928	.012227 .0395	.004931 .00645 .1076	
4	.034686	.008943 .0295		
5	.027842	.0070 .023		
6	.023242			
7	.019946			
8	.017465			

TABLE 2
Optimal-Knot Spline Approximations
to $f(x) = \sqrt{x}$ on $[0,1]$.
The first entry in each box is the minimax error magnitude. Subsequent entries give the optimal knot locations.

4	.00002716	.00000255 .5258	.00000052 .388 .660	.7507 .00000016 .308 .527 .736	.6222,.7979	.6836,.8302
5	.00000113	.00000009 .523	.00000002 .41 .67			
6	.00000004					

TABLE 1
Optimal-Knot Spline Approximations
to $f(x) = e^x$ on $[0,1]$.
The first entry in each box is the minimax error magnitude. Subsequent entries give the optimal knot locations.

4.2 Free knots. Let $\mathcal{S}_{n,k}$ be the class of splines of degree n with some k knots (allowing multiplicities) defined in §3.2.

Problem 4.2 Given $f \in C[a,b]$ determine $s^* \in \mathcal{S}_{n,k}$ such that

$$\|f - s^*\|_2 = \int_a^b [f(t) - s^*(t)]^2 \, dt \leq \|f - s\|_2$$

for every $s \in \mathcal{S}_{n,k}$.

As was the case for the uniform norm (§3.2), this is a nonlinear approximation problem which, however, does admit of a solution; see [8] where de Boor and Rice have carried out experiments on an algorithm which finds a solution with fixed knots and then adjusts them and iterates. Details and a discussion of their experience can be found in [8].

REFERENCES

1. P. M. Anselone and P. J. Laurent, A general method for the construction of interpolating or smoothing spline functions, MRC Tech. Summ. Rpt. #834, Mathematics Research Center, U. S. Army, University of Wisconsin, Madison, Wisconsin, 1967.
2. M. Attéia, Théorie et applications des fonctions-spline en analyse numérique, Thèse, Grenoble, 1966.
3. C. Carasso, Méthodes numériques pour l'obtention de fonctions-spline, Thèse de 3ème Cycle, Université de Grenoble, 1966.
4. C. Carasso, Construction numerique de fonctions-spline, 5 ème Congrès de l'Alfiro, Lille, 27 juin-1 juillet, 1966.
5. C. Carasso and P. J. Laurent, On the numerical construction and the practical use of interpolating spline functions, Proc. IFIP Congress, Edinburgh, 1968, pp. A6-A9 .
6. H. B. Curry and I. J. Schoenberg, On Pólya frequency functions. IV. The fundamental spline functions and their limits, J. d'Analyse Math. 17 (1966), 71-107.

7. C. de Boor and J. R. Rice, Least square cubic spline approximation I: Fixed knots, Rpt. CSD TR 20, Computer Science Department, Purdue University, 1968.

8. C. de Boor and J. R. Rice, Least square cubic spline approximation II: Variable knots, Rpt. CSD TR 21, Computer Science Department, Purdue University, 1968.

9. R. E. Esch and W. L. Eastman, Computational methods for best approximation, Tech. Rpt. SEG-TR-67-30, Sperry Rand Research Center, Sudbury, Massachusetts, 1967.

10. R. E. Esch and W. L. Eastman, Computational methods for best approximation and associated numerical analyses, Tech. Rpt. SRRC-CR-68-18, Sperry Rand Research Center, Sudbury, Massachusetts, 1968.

11. T. N. E. Greville, Numerical procedures for interpolation by spline functions, J. SIAM Numer. Anal. Ser. B 1 (1964), 53-68.

12. T. N. E. Greville, Interpolation by generalized spline functions, MRC Tech. Summ. Rpt. #476, Mathematics Research Center, U. S. Army, University of Wisconsin, Madison, Wisconsin, 1964.

13. T. N. E. Greville, Table for third degree spline interpolation with equally spaced arguments, MRC Tech. Summ. Rpt. #697, Mathematics Research Center, U. S. Army, University of Wisconsin, Madison, Wisconsin, 1966.

14. T. N. E. Greville, Data fitting by spline functions, Trans. 12th Conf. Army Mathematicians, Rpt. 67-1, U. S. Army Research Office-Durham, Durham, N. C., 1967, pp. 65-90; also available as MRC Tech. Summ. Rpt. # 893, Mathematics Research Center, U. S. Army, University of Wisconsin, 1968.

15. T. N. E. Greville, Introduction to spline functions, this volume, pp. 1-35.

16. J. W. Jerome and L. L. Schumaker, A note on obtaining the natural interpolating spline functions by the abstract approach of Laurent, MRC Tech. Summ. Rpt. #776,

Mathematics Research Center, U. S. Army, University of Wisconsin, Madison, Wisconsin, 1967; also to appear in SIAM J. Numer. Anal.

17. J. W. Jerome and L. L. Schumaker, On Lg-splines, to appear in J. Approximation Theory; also available as MRC Tech. Summ. Rpt. #887, Mathematics Research Center, U. S. Army, University of Wisconsin, Madison, Wisconsin, 1968.

18. C. H. Reinsch, Smoothing by spline functions, Numer. Math. 10 (1967), 177-183.

19. Sauer and Szabó, Mathematische Hilfsmittel des Ingenieurs, Springer-Verlag, Berlin, 1968.

20. I. J. Schoenberg, On interpolation by spline functions and its minimal properties, On Approximation Theory (Proceedings of the Conference held in the Mathematical Research Institute at Oberwolfach, Black Forest, Aug. 4-10, 1963), P. L. Butzer, ed., Birkhäuser Verlag, Basel, 1964, pp. 109-129.

21. L. L. Schumaker, On computing best spline approximations, MRC Tech. Summ. Rpt. #833, Mathematics Research Center, U. S. Army, University of Wisconsin, Madison, Wisconsin, 1968.

22. L. L. Schumaker, Approximation by splines, this volume, pp. 65-85.